

GUI Map loading in 2 player mods.

Starting a 2 player game with the GUI

A DOOM Eternal mod can be optimized for 2 players. However, the ready up system will only work with 3 players in the lobby, so an alternative method of loading the maps is needed. The **loaddevmenuoption** console command can forcefully load a multiplayer mod even with 1 or 2 players. Since many players will not want to enter commands, this guide will outline a more convenient graphical method.

Modifying the lobby menu to set a cvar.

HUD and menu modding in DOOM Eternal is very limited. While executing commands from a SWF is not known to be possible, there is an ActionScript function in Eternal's flash interpreter for setting Console Variables. Here's how to do this with Free Flash Decompiler:

- Open **swf/hud/menus/battle_arena/lobby_options.swf** as extracted from idStudio, or downloaded from a template version with the code for you.
- In the objects menu, navigate to scripts, item_19.
- Click "Edit ActionScript" to enable the code editor
- Scroll down to **public function item_19()** and ensure that **39,this.frame40**, is present in **addFrameScript** to ensure that the code gets executed. Some Flash editing software may add this automatically, but ffdec seems to not do this. **1,this.frame2**, and **77,this.frame78**, will also be needed for the two-column code.
- Add the following frame41 code block if it's not already present. Add the maps present in your mod, and give them a number to be used in the logic entity step later on in this guide.
- You may also wish to add the code blocks for frames 2 and 78. This code will split your map list into two columns, which may be optional or mandatory depending on how many maps are in your mod. If a mod contains too many levels, the list could get cut off the screen, so this code will make much better use of the available space. The line of code **this.wrapIndex = 8**; defines how many levels should be listed in the left column, with 0 being the first entry. So if, for example, you wanted the left column to contain only all 9 Battlemode maps, you would set this value to 8.
- Click Save in the code editor to apply the code, then click save on the ribbon menu to save the entire SWF file. A separate guide is linked below for using modded SWFs with the mod injector.

Mandatory - code for cvar output.

```
internal function frame41() : *
{
    if(this.label.txtVal.text == "Forsaken")
    {
        setCVarInteger("g_demonSelect",1);
    }
    if(this.label.txtVal.text == "Penance")
    {
        setCVarInteger("g_demonSelect",2);
    }
    if(this.label.txtVal.text == "Tundra")
    {
        setCVarInteger("g_demonSelect",3);
    }
}
```



The code will detect the string's contents, not the identifier. Either modify the strings referenced by the map info decls such that the maps have the same name consistently across localizations, or add individual translations to the ActionScript code. If you neglect this step, the code will not be effective for players that run the game in a language other than the one your multiplayer mod was tested with.

g_demonSelect was chosen because it does not have an impact on anything in the game engine, it appears to have only been added to be checked by the logic entity in the demon tutorial, which a player is not likely to enter while a multiplayer mod is installed.

Optional - code to split the map list into two columns.

```
internal function frame2() : *
{
    this.itemIndex = strReplace(this.name,"item","") * 1;
    this.wrapIndex = 8;
    if(this.itemIndex > this.wrapIndex)
    {
        if(this.x == 0)
        {
            this.x = 900;
        }
    }
}
```

```

        this.y -= 110 * (this.wrapIndex + 1);
    }
}
}

```

```

internal function frame78() : *
{
    this.itemIndex = strReplace(this.name,"item","") * 1;
    this.wrapIndex = 8;
    if(this.itemIndex > this.wrapIndex)
    {
        if(this.x == 0)
        {
            this.x = 900;
            this.y -= 110 * (this.wrapIndex + 1);
        }
    }
}
}

```

Editing the Logic Entity.

In the shell entities file, add an `idTarget_Command` entity for each map to load using the

loaddevmenuoption command. You should also add an **idTarget_Command executing any**

variables that optimize your mod if you have not done so already. Append **g_demonSelect 0** to it such that a map loading command does not get executed prematurely when the players exit a map back to the lobby menu. In the logic entity editor, add the following nodes to execute these command entities:

- A `getConsoleVariable` node to retrieve the integer value of **g_demonSelect** set through the SWF code outlined earlier.
- A begin node that connects to an activate entities node to trigger your aforementioned console command entity containing your mod's optimizations.
- The logic entity has not been tested without the delay node, but the delay is probably necessary to prevent the game from locking up because the logic entity is intentionally looped indefinitely. It's recommend to set this to around 500 milliseconds or so for optimization while keeping the map selection responsive. The node activating your optimization command entity should trigger this node.

- An integer switch function: this is what actually acts upon the demon select cvar. It should be triggered by the delay node. Make sure that the 0 option loops back to the delay node, so that the check can keep running until the player selects an option.
- activate entity nodes to connect to each individual case, each listing an idTarget_Command entity for loading a map.

Optional - add a "random map" entry.

If your mod contains several levels, you can add options for selecting a random map, just for fun. In the logic entity editor's properties dialog, add a variable of the type **Entity/List**, and add all the target command entities. If desired, you can add multiple variables, for example, a separate option that picks only the Battlemode maps. Add your node for each custom variable you added. Connect your custom variable node to a get random entity node, then connect it's entity pin to another activate entity node, which is triggered by the corresponding integer that was added for the random option.

See also

- [SWF modding](#)

Revision #3

Created 8 September 2026 00:43:05 by Alby

Updated 12 September 2026 09:08:31 by Alby